

AMENDMENTS TO THE CLAIMS

Please amend claims 1, 3, 12, 14, 16, and 25, as follows:

1. (Currently Amended) A method of optimizing a software program for a target processor to meet performance objectives, where the software program is coded in a high-level language, the method comprising the steps of:

(a) optimizing the software program such that a resulting first optimized form of the software program is substantially at least partially independent of the target processor and is substantially at least partially coded in the high-level language;

(b) optimizing the first optimized form of the software program such that a resulting second optimized form of the software program includes at least one portion that is substantially dependent on the target processor and is substantially coded in the high-level language; and

(c) flagging the substantially dependent high-level language portion of the second optimized form; and at least one portion to indicate that the at least one portion is dependent on the target processor

~~(e) — optimizing the second optimized form of the software program such that a resulting third optimized form of the software program is substantially dependent on the target processor and includes portions coded in a low-level language of the target processor.~~

2. (Original) The method of claim 1, further comprising steps of:

(a1) determining a first performance profile for the first optimized form of the software program, and comparing the first performance profile with the performance objectives; and

(b1) determining a second performance profile for the second optimized form of the software program, and comparing the second performance profile with the performance objectives.

3. (Currently Amended) The method of claim 2, ~~wherein steps (b), (b1), and (c) are not performed if the performance objectives are met after completing step (a), and step (c) is not performed if the performance objectives are met after completing step (b)~~ 1, further comprising:

(d) optimizing the second optimized form of the software program such that a resulting third optimized form of the software program is at least partially dependent on the target processor and includes portions coded in a low-level language of the target processor.

4. (Previously Presented) The method of claim 1 in which step (a) comprises the act of deriving a floating point implementation.

5. (Previously Presented) The method of claim 1 in which step (a) comprises the act of deriving a fixed point implementation.

6. (Previously Presented) The method of claim 5 in which the act of deriving the fixed point implementation comprises the act of processing qualification.

7. (Previously Presented) The method of claim 5 in which the act of deriving the fixed point implementation comprises the act of implementation sizing.

8. (Previously Presented) The method of claim 1 in which step (a) comprises the act of implementing reference code.
9. (Previously Presented) The method of claim 8 in which the act of implementing reference code comprises code profiling.
10. (Previously Presented) The method of claim 1 in which step (b) comprises the act of optimization predicted to improve resulting assembly code.
11. (Previously Presented) The method of claim 1 in which step (b) comprises the act of tuning low-level functions.
12. (Currently Amended) The method of claim ~~1~~ 3 in which step (~~e~~ d) comprises the act of manual assembly optimization.
13. (Previously Presented) The method of claim 1 in which step (b) comprises the act of feature tuning.
14. (Currently Amended) A computer-readable medium comprising a sequence of instructions which, when executed by a processor, causes the processor to execute a method for optimizing a software program for a target processor to meet performance objectives, where the software program is coded in a high-level language, the method comprising the steps of:

(a) optimizing the software program such that a resulting first optimized form of the software program is substantially at least partially independent of the target processor and is substantially at least partially coded in the high-level language;

(b) optimizing the first optimized form of the software program such that a resulting second optimized form of the software program includes at least one portion that is substantially dependent on the target processor and is substantially coded in the high-level language; and

(c) flagging the substantially dependent high-level language portion of the second optimized form; and at least one portion to indicate that the at least one portion is dependent on the target processor

~~(e) — optimizing the second optimized form of the software program such that a resulting third optimized form of the software program is substantially dependent on the target processor and is includes portions coded in a low-level language of the target processor.~~

15. (Previously Presented) The computer-readable medium of claim 14, in which the method further comprises the steps of:

(a1) determining a first performance profile for the first optimized form of the software program, and comparing the first performance profile with the performance objectives; and

(b1) determining a second performance profile for the second optimized form of the software program, and comparing the second performance profile with the performance objectives.

16. (Currently Amended) The computer-readable medium of claim 15, ~~wherein steps (b), (b1), and (c) are not performed if the performance objectives are met after completing step (a), and step (c) is not performed if the performance objectives are met after completing step (b)~~ 14, wherein the method further comprises:

(d) optimizing the second optimized form of the software program such that a resulting third optimized form of the software program is at least partially dependent on the target processor and is includes portions coded in a low-level language of the target processor.

17. (Previously Presented) The computer-readable medium of claim 14 in which step (a) comprises the act of deriving a floating point implementation.

18. (Previously Presented) The computer-readable medium of claim 14 in which step (a) comprises the act of deriving a fixed point implementation.

19. (Previously Presented) The computer-readable medium of claim 18 in which the act of deriving the fixed point implementation comprises the act of processing qualification.

20. (Previously Presented) The computer-readable medium of claim 18 in which the act of deriving the fixed point implementation comprises the act of implementation sizing.

21. (Previously Presented) The computer-readable medium of claim 14 in which step (a) comprises the act of implementing reference code.

22. (Previously Presented) The computer-readable medium of claim 21 in which the act of implementing reference code comprises code profiling.
23. (Previously Presented) The computer-readable medium of claim 14 in which step (b) comprises the act of optimization predicted to improve resulting assembly code.
24. (Previously Presented) The computer-readable medium of claim 14 in which step (b) comprises the act of tuning low-level functions.
25. (Currently Amended) The computer-readable medium of claim ~~14~~ 16 in which step (e d) comprises the act of manual assembly optimization.
26. (Previously Presented) The computer-readable medium of claim 14 in which step (b) comprises the act of feature tuning.